# UNITED STATES PATENT AND TRADEMARK OFFICE

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/643,331 | 08/19/2003 | Scott S. McDaniel | 13940US02 | 3987 |

23446          7590          11/13/2009
MCANDREWS HELD & MALLOY, LTD
500 WEST MADISON STREET
SUITE 3400
CHICAGO, IL 60661

| EXAMINER |
|---|
| BELANI, KISHIN G |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2443 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 11/13/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

|  | Application No. | Applicant(s) |
|---|---|---|
| **Office Action Summary** | 10/643,331 | MCDANIEL ET AL. |
|  | Examiner | Art Unit |
|  | KISHIN G. BELANI | 2443 |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on _27 August 2009_.

2a)☐ This action is **FINAL**.    2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) _1-25_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-25_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All    b)☐ Some * c)☐ None of:

        1.☐ Certified copies of the priority documents have been received.

        2.☐ Certified copies of the priority documents have been received in Application No. _____.

        3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

## DETAILED ACTION

This action is in response to Applicant's RCE filed on 08/27/2009. **Independent claims 1, 17, 24, 25 and dependent claims 2-5, 10, 11, and 21 have been amended**. The applicants' amendments to claims are shown in ***bold and italics***, and the examiner's response to the claim amendments is shown in **bold** in this office action. **Claims 1-25** are now pending in the present application.

### *Continued Examination Under 37 CFR 1.114*

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 08/27/2009 has been entered.

### *Claim Rejections - 35 USC § 102*

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

**Claims 1-5, 10, and 16** are rejected under 35 U.S.C. 102(b) as being anticipated

by **Osborne (U.S. Patent Publication # 5,790,804).**

Consider **claim 1**, Osborne shows and discloses a system for transferring data

over a remote direct memory access (RDMA) network (Figs. 1 and 2 that show a

system with an application 58, at the sender 50 and receiver 52 communicate by

sending messages to each other across the network 82; column 7, lines 14-45 disclose

the details of the system, including reciting that the operating system may set up a

direct memory access (DMA) with the network interface to extract and copy the

message sent from sender's application 1 to message buffer 74 of the receiver's

application 1); comprising:

a host comprising a driver and a network interface card (NIC), the driver being coupled

to the NIC (Fig. 1 that shows a host (sender 50) with operating system 56, processor

(driver) 54, and a network interface 84 that is coupled to the processor; column 7, lines

14-45 disclose the details of the system);

wherein a one-shot initiation process of an RDMA operation is performed between the

driver and the NIC of the host (Figs. 1-2 that show the system and list the steps

necessary (in Fig. 2) to initiate, by the processor 54 and NIC 84 of host 50, the transfer

of a message data from application 1 to the corresponding application 1 at the receiving

host 52; column 7, lines 18-36 describe the steps shown in Fig. 2, starting at step 99, by

invoking a "Send" command 67 (details shown in Fig. 1) by application 1, which is

copied over from the application memory to processor message buffer 62 in step 100,

and processing, by processor 54, the message content to a format compatible with the

protocol being used in step 102, before handing over the initiation process to NIC 84,

that sends the message and its content over the network 82 in step 104, for delivery to

a remote receiving host 52; column 7, lines 37-45 further disclose using a direct

memory access (DMA) with the network interface to extract and copy the message to

message buffer 74 of the remote host 52);

the one-shot initiation process comprising communicating a single command message

comprising:

buffer command information (Fig. 1, "Send" command 67 with parameters ID (identifying

the receiver of the message), Source Address (location where the message content is

stored), and Size (amount of data to be sent) stored in the buffer 66; column 7, lines 19-

25 disclose the same details), and

a write command to write a send command (Fig. 2, step 100 that shows a copy (write)

command to copy message data from application memory to processor's message

buffer 62 (see Fig. 1)).


        Consider **claim 2**, and **as it applies to claim 1 above**, Osborne further shows

and discloses the claimed system wherein the driver posts the single command

message to perform the one-shot initiation process (Fig. 1, processor (driver) 54 that

posts the Send(ID, Source Address, Size) command 67 to Network Interface 84, by

copying the command from application 1 memory 66 to message buffers 62-64 of the

processor's operating system 56; Fig. 2, steps 100-104 that show the details of posting

the send message to the NIC 84; column 7, lines 14-45 disclose the same details).

Consider **claim 3**, and **as it applies to claim 1 above**, Osborne further shows

and discloses the claimed system wherein the buffer command information comprises a

command to describe pinned-down memory buffers of the host (Fig. 1, pinned-down

message buffers 62-64 of sender (host) 50 that store the Send command with its

parameters (such as Source Address that points to the message's content) and the

message content copied from the Application 1 memory 66; column 7, lines 25-31 teach

the same details, further disclosing that the copying step is often optimized by mapping

locations in the application memory 66 to locations in the message buffer 62 to avoid

actual copying).

Consider **claim 4**, and **as it applies to claim 3 above**, Osborne further discloses

the claimed system wherein the buffer command information comprises a command to

bind a portion of the pinned down memory buffers of the host to a steering tag (STag)

(column 7, lines 25-31 which further disclose that the copying step is often optimized by

mapping locations (using pointers, considered by the examiner to correspond to

applicants' steering tag STag, to de-reference the memory location) where the message

content (i.e. payload) is actually stored in the application memory 66 to locations in the

message buffer 62, so as to avoid actual copying of the content in the message buffers

62-64).

Consider **claim 5**, and **as it applies to claim 1 above**, Osborne further discloses

the claimed system wherein the send command is an RMDA send message (Fig. 1,

"Send" command 67 with parameters ID (identifying the receiver of the message),

Source Address (location where the message content is stored), and Size (amount of

data to be sent) stored in the buffer 66; column 7, lines 19-25 disclose the same details;

and column 7, lines 37-45 which disclose using a direct memory access (DMA) with the

network interface to send the message to message buffer 74 of the remote host 52).

Consider **claim 10**, and **as it applies to claim 1 above**, Osborne shows and

discloses the claimed system, wherein the buffer command information provides a

description of a section of memory (Fig. 1, Send command 67 with the parameter

"Source Address" that points to a memory location in the application memory 66, where

the content of the message being sent by RDMA is stored, thereby providing a

description of a section of memory; the corresponding content when copied into the

message buffers 62-64 of the operating system 56, also show that the buffer command

information provides a description of a section of memory; column 7, lines 14-45

describe the same details).

Consider **claim 16**, and **as it applies to claim 1 above**, Osborne shows and

discloses the claimed system wherein the NIC comprises an RDMA-enabled NIC (Fig. 1

that shows Network Interface (NIC) 84; column 7, lines 37-45 describe the NIC as a

Remote DMA-enabled NIC).

## *Claim Rejections - 35 USC § 103*

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148

USPQ 459 (1966), that are applied for establishing a background for determining

obviousness under 35 U.S.C. 103(a) are summarized as follows:

1.   Determining the scope and contents of the prior art.
2.   Ascertaining the differences between the prior art and the claims at issue.
3.   Resolving the level of ordinary skill in the pertinent art.
4.   Considering objective evidence present in the application indicating obviousness
     or nonobviousness.

This application currently names joint inventors. In considering patentability of

the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of

the various claims was commonly owned at the time any inventions covered therein

were made absent any evidence to the contrary. Applicant is advised of the obligation

under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was

not commonly owned at the time a later invention was made in order for the examiner to

consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g)

prior art under 35 U.S.C. 103(a).


**Claims 6-9, 11 and 24** are rejected under 35 U.S.C. 103(a) as being

unpatentable over **Osborne (U.S. Patent Publication # 5,790,804)** in view of **Roach et**

**al. (U.S. Patent Publication # 6,304,910 B1).**


Consider **claim 6**, and **as it applies to claim 4 above**, Osborne discloses the

claimed system, except wherein the NIC places the STag value in an optional field in a

direct data placement DDP or RDMA header.

In the same field of endeavor, Roach et al. show and disclose the claimed

system wherein the NIC places the STag value in an optional field in a direct data

placement DDP or RDMA header (Fig. 8 that shows the bit-level definition of each of the

two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18

that disclose the details of the bit-level structure including BSWA buffer pointer list

entries corresponding to the STag values and BLM flag fields indicating validity of the

BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to place the STag value in an optional field in a direct

data placement DDP or RDMA header, as taught by Roach et al., in the system of

Osborne, so as to be able to handle non-contiguous data blocks during the transfer.

Consider **claim 7**, and **as it applies to claim 6 above**, Osborne discloses the claimed system, except wherein the NIC encodes a value into a field in the DDP or RDMA header indicating that the STag value in the optional field is valid.

In the same field of endeavor, Roach et al. show and disclose a system wherein the NIC encodes a value into a field in the DDP or RDMA header indicating that the STag value in the optional field is valid (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list entries corresponding to the STag values and BLM flag fields indicating validity of the BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to encode a value into a field in the DDP or RDMA header indicating that the STag value in the optional field is valid, as taught by Roach et al., in the system of Osborne, so as to be able to distinguish valid data block entries for remote transfer.

Consider **claim 8**, and **as it applies to claim 6 above**, Osborne discloses the claimed system, except wherein the NIC sets one or more bits in a field in the DDP or RDMA header indicating that the STag value in the optional field is valid.

In the same field of endeavor, Roach et al. show and disclose a system wherein the NIC sets one or more bits in a field in the DDP or RDMA header indicating that the STag value in the optional field is valid (Fig. 8 that shows the bit-level definition of each

of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines

1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list

entries corresponding to the STag values and BLM flag fields indicating validity of the

BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to set a value into a field in the DDP or RDMA header

indicating that the STag value in the optional field is valid, as taught by Roach et al., in

the system of Osborne, so as to be able to distinguish valid data block entries for

remote transfer.

Consider **claim 9**, and **as it applies to claim 6 above**, Osborne discloses the

claimed system, except wherein the NIC sets one or more bits or encodes a value into a

second field in the DDP or RDMA header to advertise the portion of the pinned memory

buffers of the host associated with the STag.

In the same field of endeavor, Roach et al. show and disclose a system wherein

the NIC sets one or more bits or encodes a value into a second field in the DDP or

RDMA header to advertise the portion of the pinned memory buffers of the host

associated with the STag (Fig. 8 that shows the bit-level definition of each of the two 32-

bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that

disclose the details of the bit-level structure including BSWA buffer pointer list entries

corresponding to the STag values and BLM flag fields indicating validity of the BSWA

entries, thereby advertising the portion of the pinned memory buffers of the host

associated with the STag).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to set one or more bits or encode a value into a

second field in the DDP or RDMA header to advertise the portion of the pinned memory

buffers of the host associated with the STag, as taught by Roach et al., in the system of

Osborne, so as to be able to distinguish valid data block entries for remote transfer.

Consider **claim 11**, and **as it applies to claim 1 above**, Osborne discloses the

claimed system, except wherein the single command message is posted to a command

ring of the host.

In the same field of endeavor, Roach et al., disclose a system wherein the single

command message is posted to a command ring of the host (Fig. 5A, block 32; column

6, lines 48-64 that disclose a command ring structure as a circular queue of command

entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to post the single command message to a command

ring of the host, as taught by Roach et al., in the system of Osborne, so as to be able to

execute the commands in the FIFO order with opportunity for the non-serviced

commands to get multiple turns at the service.

Consider **claim 24**, Osborne shows and discloses a method for transferring data

over an RDMA network **(Figs. 1 and 2 that show a method used with an application**

**58, at the sender 50 and receiver 52 wherein the sender and receiver**

**communicate by sending messages to each other across the network 82; column**

**7, lines 14-45 disclose the details of the method, including reciting that the**

**operating system may set up a direct memory access (DMA) with the network**

**interface to extract and copy the message sent from sender's application 1 to**

**message buffer 74 of the receiver's application 1)**, comprising:

initiating an RDMA write operation using a one-shot initiation process between a driver

and a NIC of a host **(Fig. 1 that shows a host (sender 50) with operating system 56,**

**processor (driver) 54, and a network interface 84 that is coupled to the processor;**

**column 7, lines 14-45 disclose the details of the method; Figs. 1-2 that further**

**show the method and list the steps necessary (in Fig. 2) to initiate, by the**

**processor 54 and NIC 84 of host 50, the transfer of a message data from**

**application 1 to the corresponding application 1 at the receiving host 52; column**

**7, lines 18-36 describe the steps shown in Fig. 2, starting at step 99, by invoking a**

**"Send" command 67 (details shown in Fig. 1) by application 1, which is copied**

**over from the application memory to processor message buffer 62 in step 100,**

**and processing, by processor 54, the message content to a format compatible**

**with the protocol being used in step 102, before handing over the initiation**

**process to NIC 84, that sends the message and its content over the network 82 in**

**step 104, for delivery to a remote receiving host 52; column 7, lines 37-45 further**

**disclose using a direct memory access (DMA) with the network interface to**

**extract and copy the message to message buffer 74 of the remote host 52);**

*wherein the one-shot initiation process comprises communicating a single*

*command message comprising buffer command information comprising*

*commands to insert an STag value, and a write command to write an RDMA send*

*message* **(Fig. 1 that shows a one-shot initiation process that comprises**

**communicating a single Send command 67 message with parameters ID (of**

**destination host), source address pointer (STag value that is part of the buffer**

**command information), and message size (also a part of the buffer command**

**information) from a sending host 50 to a receiving host 52 via a processor 54, NIC**

**84, network 82 to NIC 86, processor 70, and finally to receiving host 52), Fig. 1**

**further showing a copying process to copy the send command and the**

**associated parameters of the send command from the Application 1 buffers 66 to**

**the operating system's message buffers 62-64, thereby disclosing a write**

**command to write an RDMA send message).**

However, Osborne does not show or disclose *validating STag value;* inserting

an STag value in a first field of a DDP or RDMA header of the RDMA send message;

and validating the STag value in the first field with a bit flag or other specified value in a

second field of the DDP or RDMA header.

In the same field of endeavor, Roach et al. show and disclose inserting an STag

value in a first field of a DDP or RDMA header of an RDMA send message (Fig. 8 that

shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9;

column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level

structure including BSWA buffer pointer list entries corresponding to the STag values);

and validating the STag value in the first field with a bit flag or other specified value in a

second field of the DDP or RDMA header (Fig. 8 that shows the bit-level definition of

each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9,

lines 1-18 that disclose the details of the bit-level structure including BLM flag fields

indicating validity of the BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to insert the STag value in a first field of a DDP or

RDMA header of an RDMA send message, and validate the STag value in the first field

with a bit flag or other specified value in a second field of the DDP or RDMA header, as

taught by Roach et al., in the method of Osborne, so as to be able to handle non-

contiguous data blocks during the transfer and distinguish valid data block entries for

remote transfer.

**Claims 12-15** are rejected under 35 U.S.C. 103(a) as being unpatentable over

**Osborne (U.S. Patent Publication # 5,790,804)** in view of **Roach et al. (U.S. Patent**

**Publication # 6,304,910 B1)** and further in view of **Tillier (U.S. Patent Publication #**

**6,421,742 B1).**

Consider **claim 12**, and **as it applies to claim 11 above**, Osborne as modified by Roach et al., discloses the claimed system, except wherein the driver allocates an STag value.

In the same field of endeavor, Tillier shows and discloses the claimed system wherein the driver allocates an STag value (Fig. 8, arrow marked 'Pointer to "A"' which the examiner has interpreted to be equivalent to a STag value, being allocated by the Input/output unit (driver) and received by NIC (functionally represented by the middle column), the pointer value representing the address of the pinned down memory).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a system wherein the driver allocates an STag value, as taught by Tillier, in the system of Osborne, as modified by Roach et al., so as to associate and locate the pinned-down memory buffers using the STag value.

Consider **claim 13**, and **as it applies to claim 12 above**, Osborne as modified by Roach et al., and Tillier, further discloses the claimed system, wherein the STag value is returned synchronously from a command call (in Roach et al. reference, column 6, lines 56-64 which disclose that the host driver increments the "put" pointer whenever a command is queued to the command ring, making the updated pointer (STag value) synchronously available from a call command).

Consider **claim 14**, and **as it applies to claim 12 above**, Osborne, as modified by Roach et al., and Tillier, further discloses the claimed system, wherein the STag

value is saved in a driver command table of the host (in Roach et al. reference, Fig. 3,

host memory block 42 and transfer ready queue 60; column 5, lines 44-52 that disclose

a lookup field (STag value) inside each frame header includes a pointer to an

associated context, and the host driver saves these fields in the host memory 42).

.

Consider **claim 15**, and **as it applies to claim 14 above**, Osborne, as modified

by Roach et al., and Tillier, further discloses the claimed system, wherein the STag

value saved in a driver command table is associated with an application reference

number (in Roach et al. reference, Figs. 5A and 5B; column 5, lines 47-52 which

disclose that the context field associated with a pointer field (STag value) saved in a

driver command table is associated with a small computer systems interface (SCSI)

state information interpreted by the examiner to be associated with the application

reference number).

**Claim 17** is rejected under 35 U.S.C. 103(a) as being unpatentable over

**Osborne (U.S. Patent Publication # 5,790,804)** in view of **Pandya (U.S. Patent**

**Publication # 7,376,755 B2).**

Consider **claim 17**, Osborne shows and discloses a system for transferring data

over a remote direct memory access (RDMA) network **(Figs. 1 and 2 that show a**

**system with an application 78, at the receiver host 52, that communicates with the**

**corresponding application 1 at the sender host 50, by receiving/sending**

messages to each other across the network 82; column 7, lines 14-45 disclose the

details of the system, including reciting that the operating system may set up a

direct memory access (DMA) with the network interface to extract and copy the

message sent from sender's application 1 to message buffer 74 of the receiver's

application 1), comprising:

a host comprising a driver and a network interface card (NIC), the driver being

coupled to the NIC (Fig. 1 that shows a receiving host 52 with operating system

72, processor (driver) 70, and a network interface 86 that is coupled to the

processor; column 7, lines 14-45 disclose the details of the system),

wherein a one-shot completion process of an RDMA operation is performed between

the driver and the NIC of the host (**Figs. 1-2 that show the system and list the steps**

**necessary (in Fig. 2) to complete, by the processor 70 and NIC 86 of the receiving**

**host 52, the transfer of a message data from application 1 at the sender host 50 to**

**the corresponding application 1 at the receiving host 52, and a "Receive"**

**command 77 (details shown in Fig. 1); column 7, lines 37-65 describe the**

**message receiving steps 106-110 shown in Fig. 2, further disclosing that the**

**message arrival at the receiver causes an interrupt to the processor 70, and the**

**operating system 72 extracts the message (step 106) from the network in**

**cooperation with NIC 86, and stores it into message buffer 74 in the operating**

**system using Direct Memory Access (DMA); the operating system 72 then**

**performs protocol processing, if necessary, in step 108; then copies the message**

**in step 110 from the message buffer 72 at the receiver 52 to application memory**

**(endpoint 79), in response to a "Receive" command request 77 from application 78).**

However, Osborne does not explicitly disclose that the one-shot completion process comprises communicating a single completion message comprising:

a send complete indication, and buffer freeing status information.

In the same field of endeavor, Pandya discloses the claimed system, including sending a send complete indication, and buffer freeing status information (Fig. 33 that shows the details of a "Write" operation, including send status and sense step 3310 at the completion of the "Send Data" operation; column 13, lines 35-48 which describe that at the completion of data transfer (associated with the send data command), the status of the command execution is passed onto the host SCSI layer, which then does the appropriate processing, including releasing the buffers being used for data transfer to the application).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to include a send complete indication, and buffer freeing status information, as taught by Pandya, in the system of Osborne, so as to enable the sending host to perform clean-up operations after the message with its content has been successfully received by the receiving host.


**Claims 18, 20, 22 and 23** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Osborne (U.S. Patent Publication # 5,790,804)** in view of **Pandya**

**(U.S. Patent Publication # 7,376,755 B2)** and further in view of **Tillier (U.S. Patent**

**Publication # 6,421,742 B1).**


Consider **claim 18**, and **as it applies to claim 17 above**, Osborne, as modified

by Pandya, shows and discloses the claimed system, except wherein the NIC receives

a message comprising an optional field carrying a STag value, the STag value being

associated with pinned memory in a remote host.

In the same field of endeavor, Tillier shows and discloses the claimed system,

wherein the NIC receives a message comprising an optional field carrying an STag

value, the STag value being associated with pinned memory in a remote host (Fig. 8,

arrow marked 'Pointer to "A"' which the examiner has interpreted to be equivalent to a

STag value, being received by NIC along with the "Store Data" command, the pointer

value representing the address of the pinned down memory in a remote host).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a system wherein the NIC receives a

message comprising an optional field carrying an STag value, the STag value being

associated with pinned memory in a remote host, as taught by Tillier, in the system of

Osborne, as modified by Pandya, so as to be able to access the message content in the

pinned memory in the remote host.


Consider **claim 20**, and **as it applies to claim 18 above**, Osborne, as modified

by Pandya and Tillier, further shows and discloses a system wherein the NIC de-

associates the STag value with the pinned memory in the host, thereby preventing

further access to the pinned memory using the de-associated STag value (in Tillier

reference, Fig. 6, blocks 604-606 that depict freeing up of allocated resources,

corresponding to de-associating the STag value from the pinned-memory in the host,

thereby preventing further access to the pinned memory using the de-associated STag

value; column 8, lines 9-15 that disclose the same details).

Consider **claim 22**, and **as it applies to claim 18 above**, Osborne, as modified

by Pandya and Tillier, further shows and discloses the claimed system wherein the NIC

de-associates the STag value with previously associated SGL information (in Tillier

reference, Fig. 6, blocks 604-606 that depict freeing up of allocated resources,

corresponding to de-associating the STag value pointing to the SGL; cleanup routine of

Fig. 7B; column 8, lines 9-15 that disclose the same details).

Consider **claim 23**, and **as it applies to claim 20 above**, Osborne, as modified

by Pandya and Tillier, further shows and discloses a system wherein the NIC frees any

resources dedicated to information regarding the pinned memory (in Tillier reference,

Fig. 6, blocks 604-606 that depict freeing up of allocated resources; cleanup routine of

Fig. 7B; column 8, lines 9-15 that disclose NIC freeing up the allocated resources).

**Claim 19** is rejected under 35 U.S.C. 103(a) as being unpatentable over

**Osborne (U.S. Patent Publication # 5,790,804)** in view of **Pandya (U.S. Patent**

**Publication # 7,376,755 B2)** and further in view of **Tillier (U.S. Patent Publication #**

**6,421,742 B1)** and further in view of **Roach et al. (U.S. Patent Publication # 6,304,910**

**B1).**

Consider **claim 19**, and **as it applies to claim 18 above**, Osborne, as modified

by Pandya and Tillier, discloses the claimed system, except wherein a header of the

message indicates the validity of the optional field with a bit flag or specified value in an

encoded field.

In the same field of endeavor, Roach et al. show and disclose a system wherein

a header of the message indicates the validity of the optional field with a bit flag or

specified value in an encoded field (Fig. 8 that shows the bit-level definition of each of

the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-

18 that disclose the details of the bit-level structure including BSWA buffer pointer list

entries corresponding to the STag values and BLM flag fields indicating validity of the

BSWA entries, thereby indicating the validity of the optional field with a bit flag or

specified value in an encoded field).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to indicate the validity of the optional field with a bit flag

or specified value in an encoded field, as taught by Roach et al., in the system of

Osborne, as modified by Pandya and Tillier, so as to be able to distinguish valid data

block entries for remote transfer.

Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over

**Osborne (U.S. Patent Publication # 5,790,804)** in view of **Pandya (U.S. Patent**

**Publication # 7,376,755 B2)** and further in view of **Tillier (U.S. Patent Publication #**

**6,421,742 B1)** and further in view of **Futral et al. (U.S. Patent Publication #**

**5,991,797).**

Consider **claim 21**, and **as it applies to claim 18 above**, Osborne, as modified

by Pandya and Tillier, further discloses the claimed system, ***wherein the single***

***completion message comprises the optional field carrying the STag value***

***received by the NIC;*** and wherein the NIC delivers the ***single completion*** message to

the driver **(in Pandya reference, Fig. 35, Target Operations 3507-3509 that send**

**"Status and Sense" message via NIC and Driver to the Initiator of "Write"**

**command at the completion of the command; since the STag field is optional, it is**

**not shown; the NIC then delivers the message to the LAN driver 716 shown in Fig.**

**7; column 34, lines 29-30 disclose the response to data transfer completion**

**operation).**

However, Osborne, as modified by Pandya and Tillier, do not specifically show or

disclose the claimed system wherein the driver compares the STag value received with

a STag value previously sent.

In the same field of endeavor, Futral et al. show and disclose a system wherein

the NIC delivers the message to the driver, and wherein the driver compares the STag

value received with a STag value previously sent (Fig. 4, SGL word 2 (chain pointer)

that provides address to additional transaction detail element list, when the buffers are

scattered across multiple locations, therefore requiring an appended list.  In such cases,

the driver compares the SGL address received with a prior value to ensure that the

same list is being processed with additional buffer addresses in the appended chain list;

column 6, lines 36-46 discloses the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide capability in the driver to compare the STag

value received with a STag value previously sent, as taught by Futral et al., in the

system of Osborne, as modified by Pandya and Tillier, so as to be able to handle

multiple non-contiguous data blocks during the transfer.


**Claim 25** is rejected under 35 U.S.C. 103(a) as being unpatentable over

**Osborne (U.S. Patent Publication # 5,790,804)** in view of **Pandya (U.S. Patent**

**Publication # 7,376,755 B2)** and further in view of **Roach et al. (U.S. Patent**

**Publication # 6,304,910 B1).**


Consider **claim 25**, Osborne shows and discloses a method for transferring data

over an RDMA network **(Figs. 1 and 2 that show a method used with an application**

**58, at the sender 50 and receiver 52 wherein the sender and receiver**

**communicate by sending messages to each other across the network 82; column**

**7, lines 14-45 disclose the details of the method, including reciting that the**

**operating system may set up a direct memory access (DMA) with the network**

**interface to extract and copy the message sent from sender's application 1 to message buffer 74 of the receiver's application 1)**, comprising:

completing an RDMA write operation using a one-shot completion process between a NIC and a driver of a host (**Figs. 1-2 that show the system and list the steps necessary (in Fig. 2) to complete, by the processor 70 and NIC 86 of the receiving host 52, the transfer of a message data from application 1 at the sender host 50 to the corresponding application 1 at the receiving host 52, and a "Receive" command 77 (details shown in Fig. 1); column 7, lines 37-65 describe the message receiving steps 106-110 shown in Fig. 2, further disclosing that the message arrival at the receiver causes an interrupt to the processor 70, and the operating system 72 extracts the message (step 106) from the network in cooperation with NIC 86, and stores it into message buffer 74 in the operating system using Direct Memory Access (DMA); the operating system 72 then performs protocol processing, if necessary, in step 108; then copies the message in step 110 from the message buffer 72 at the receiver 52 to application memory (endpoint 79), in response to a "Receive" command request 77 from application 78**).

However, Osborne does not specifically show and disclose that *the one-shot completion process comprises communicating a single completion message comprising:*

*a send complete indication, buffer freeing status information, and an STag value;*

receiving the single completion message;

identifying the STag value in a first field of a header of the single completion message;

and validating the STag value in the first field of the header by identifying a bit flag or

other specified value in a second field of the header.

In the same field of endeavor, Pandya shows and discloses the claimed method,

wherein *the one-shot completion process comprises communicating a single*

*completion message comprising:*

*a send complete indication, buffer freeing status information, and an STag value*

**(Fig. 33 that shows the details of a "Write" operation, including send status and**

**sense step 3310 at the completion of the "Send Data" operation; column 13, lines**

**35-48 which describe that at the completion of data transfer (associated with the**

**send data command), the status of the command execution is passed onto the**

**host SCSI layer, which then does the appropriate processing, including releasing**

**the buffers being used for data transfer to the application; since releasing the**

**buffers implies knowing where in the memory the buffers are, it is implied that a**

**pointer (STag) to the message buffers is also sent with the send complete**

**indication message)***; and*

receiving *the single* completion message **(Fig. 33, message "send status and**

**sense" 3310 from the target to the initiator in response to the "Write" command**

**request 3301, wherein the message indicates command complete message 3312**

**at the initiator; column 13, lines 35-48 which describe the same details)**.

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide in the one-shot completion process that

comprises communicating a single completion message, a send complete indication,

buffer freeing status information, and an STag value, as taught by Pandya, in the

method of Osborne, so as to free up the locked (pinned-down) buffers after the data

transfer between the two hosts is complete.

However, Osborne, as modified by Pandya, does not disclose identifying a STag

value in a first field of a header of the completion message; and validating the STag

value in the first field of the header by identifying a bit flag or other specified value in a

second field of the header.

In the same field of endeavor, Roach et al. show and disclose identifying a STag

value in a first field of a header of the completion message (Fig. 8 that shows the bit-

level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-

67 and column 9, lines 1-18 that disclose the details of the bit-level structure including

BSWA buffer pointer list entries corresponding to the STag values); and

validating the STag value in the first field of the header by identifying a bit flag or other

specified value in a second field of the header (Fig. 8 that shows the bit-level definition

of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column

9, lines 1-18 that disclose the details of the bit-level structure including BLM flag fields

indicating validity of the BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to identify a STag value in a first field of a header of

the completion message, and validate the STag value in the first field of the header by

identifying a bit flag or other specified value in a second field of the header, as taught by

Roach et al., in the method of Osborne, as modified by Pandya, so as to be able to

handle non-contiguous data blocks during the transfer and distinguish valid data block

entries for remote transfer.


### *Response to Arguments*

Applicants' arguments with respect to **claims 1-25** have been considered but are

moot in view of the new ground(s) of rejection.  However, the examiner would like to

respond to some of the arguments in order to make his position clear.  The examiner's

response to applicants' arguments is given below:

Consider **independent claims 1 and 17.**  On page 9 of the "Remarks" section,

the applicants argue that since the cited Pandya reference discloses registering RDMA

buffers before issuing the read or write command (i.e. multiple steps), Pandya cannot

disclose "wherein a one-shot initiation process of an RDMA operation is performed

between the driver and the NIC of the host, the one-shot initiation process comprising

communicating a single command message comprising: buffer command information,

and a write command to write a send command", as recited in claim 1.  The examiner

would like to point out that the cited sections in the Pandya reference (Fig. 7, and

column 11, lines 10-62) teach that the purpose of registering the pinned-down memory

regions with the HBA/NIC with their access control keys along with the region IDs is to

allow the system to be secure and prevent unauthorized access (column 11, lines 46-

54).  Since the applicants' invention does not offer this claim feature in their invention,

their argument about calling the RDMA initiation of Pandya reference as a multiple step

operation is without any basis. However, the examiner has presented a new Osborne reference that more clearly shows and discloses one-shot RDMA initiation and completion processes. The examiner, however, still maintains that the prior rejection for claims 1 and 17 based on the cited Pandya reference is perfectly valid. The applicants' additional argument about Pandya's Figs. 35 and 37 showing two separate (register buffers and write) commands, is also without merit for the same reason listed above.

On page 10 of the "Remarks" section, the applicants present argument that is unrelated to claim 1 rejection. The examiner rejected claim 1 on the basis of 35 USC 102(e) as being anticipated by Pandya reference. The examiner doesn't understand the applicants' argument about the cited Tillier reference, which wasn't even used in rejecting claim 1.

On page 11 of the "Remarks" section, the applicants state that Pandya's steps 3507-3509 in Fig. 35 merely show that the target sends the initiator an indication that operation is complete, without detailing the specifics of the process, further arguing that steps 3507-3509 are not mentioned in the specification of Pandya. The examiner considers this as a very weak argument. A small detail like a completion status may merely be shown, without elaboration. However, Pandya (in column 34, lines 29-30) does disclose that the operation completion is indicated using the command completion response. As to Pandya reference not detailing freeing buffers at the end of completing the command, it is a good programming practice to always free up previously reserved buffers at the end of successful completion of a process, so that other applications have

access to those buffers. The freeing up of buffers is therefore inherently implied in

Pandya, and therefore, not specifically mentioned.

Again, on page 12 of the "Remarks" section, the applicants present argument

that is unrelated to claim 17 rejection. The examiner rejected claim 17 on the basis of

35 USC 102(e) as being anticipated by Pandya reference. The examiner doesn't

understand the applicants' argument about the cited Tillier reference, which wasn't even

used in rejecting claim 17.

On page 14 of the "Remarks" section, the applicants argue that Tillier's

disclosure regarding its emulation service layer 103-3 in its I/O device 103 is completely

unrelated to commands sent between a driver and a NIC in Tillier's host 101, therefore

Tillier's "Map buffer 506" of Fig. 5 as showing a command to bind a portion of pinned-

down memory buffers of the host to a steering tag (STag) is not equivalent. The

examiner respectfully disagrees with this argument. An emulation service mimics in

software the actions performed by hardware (e.g. a processor and a NIC), resulting in

the same outcome. The I/O device 103 emulates the Driver+NIC hardware's actions,

and the "Map buffer" action shown in Fig. 5, allocates memory buffers for emulating the

data transfer from host 101 to another host via I/O device 103, wherein the address of

the allocated memory buffers is saved in a pointer field (corresponding to "Map buffer"

operation), which is equivalent to the steering tag (STag) mentioned by the applicants,

thereby teaching binding a portion of pinned-down memory buffers of the host to a

steering tag (STag).

On page 15 of the "Remarks" section, the applicants make the same argument for dependent claim 10 that they made for claim 4, to which the examiner has already responded above. The applicants also argue about the dependent claim 18, stating that the "Pointer to A" in Tillier's Fig. 8 is not an **optional** field of the message, and is not associated with pinned memory in a **remote** host. The examiner would like to point out that by making the field optional, the applicants have permitted the examiner to assign no weight to that feature, since it may or may not be present. So, the argument is moot. The "pinned memory in a **remote** host" feature was shown and disclosed in the Pandya reference, and need not be repeated in the Tillier reference. Furthermore, the newly cited reference of Osborne clearly shows these features as well.

On page 16 of the "Remarks" section, the applicants further argue that nowhere in Tillier is there any mention of de-association and freeing of resources performed at a NIC, as recited in the applicants' claims 20, 22, and 23. The examiner begs to differ with this assertion. As stated above, Tillier describes an emulator mimicking the functions of hardware operations carried out by a processor (Driver) and NIC. Therefore, Tillier would not show such hardware, but it clearly shows and discloses the corresponding functions of de-associating the STag value from the pinned memory in the host, and freeing up the pinned-memory buffers, as shown in Fig. 6, steps 605-606, and disclosed in column 8, lines 9-15.

## *Conclusion*

Any response to this Office Action should be **faxed to** (571) 273-8300 **or mailed**

**to**:

> Commissioner for Patents
> P.O. Box 1450
> Alexandria, VA 22313-1450

Art Unit: 2443

**Hand-delivered responses** should be brought to

> Customer Service Window
> Randolph Building
> 401 Dulany Street
> Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the

Examiner should be directed to Kishin G. Belani whose telephone number is (571) 270-

1768. The Examiner can normally be reached on Monday-Friday from 6:00 am to 5:00

pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's

supervisor, Tonia Dollinger can be reached on (571) 272-4170. The fax phone number

for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for published

applications may be obtained from either Private PAIR or Public PAIR. Status

information for unpublished applications is available through Private PAIR only.  For

more information about the PAIR system, see http://pair-direct.uspto.gov. Should you

have questions on access to the Private PAIR system, contact the Electronic Business

Center (EBC) at 866-217-9197 (toll-free) or 703-305-3028.

Any inquiry of a general nature or relating to the status of this application or

proceeding should be directed to the receptionist/customer service whose telephone

number is (571) 272-0800.


*/K. G. B./*
*Examiner, Art Unit 2443*

October 29, 2009

/George C Neurauter, Jr./
Primary Examiner, Art Unit 2443